



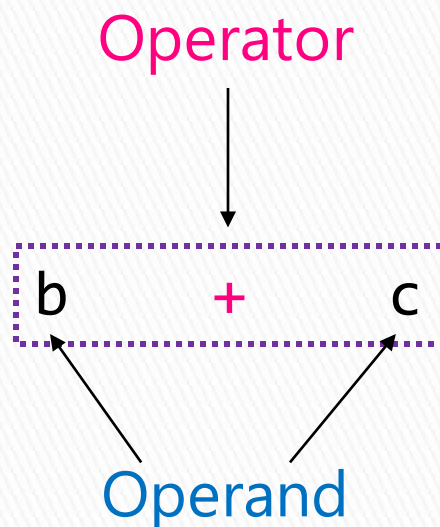
Öğr.Gör. Mustafa GÖKMEN  
gokmen@selcuk.edu.tr

# Nesneye Yönelik Programlama

Operators

# Operatörler

Operatörler programlama dillerinin en temel işlem yapma yeteneğine sahip simgesel isimlerdir. İki veya daha fazla operand arasında yapılacak eylemin türünü ifade ederler.



# Operatörler

## Aritmetik (Arithmetic) Operatörler

Operandlar arasındaki matematiksel işlemler için kullanılırlar.

Operasyon	Operatör	Örnek	Değer (a=10, b=7, c=2.5)
Toplama	+	a + b	17
Çıkarma	-	a - b	3
Çarpma	*	a * b	70
Bölme	/	a / b a / c	1 4.0
Modüler	%	a % b	3

Aritmetik operatörleri **birleşik** olarak aşağıdaki gibi gösterebiliriz.

Operatör	Kullanım	Anlamı
+=	a += b;	a = a + b;
--	a -= b;	a = a - b;
*=	a *= b;	a = a * b;
/=	a /= b;	a = a / b;
%=	a %= b;	a = a % b;

# Operatörler

## Artırma (Increment)-Azaltma (Decrement) Operatörleri

Operatör	Gösterimi	Örnek (a = 5)
Artırma (Increment)	++	<code>++a //Pre-Increment</code> 5 -> (6)
		<code>a++ //Post-Increment</code> 5 -> (6)
Azaltma (Decrement)	--	<code>--a //Pre-Decrement</code> 4 -> (4)
		<code>a-- //Post-Decrement</code> 5 -> (4)

- ❖ Pre-Increment operatörü operandı işleme sokmadan önce değerini 1 artırır.
- ❖ Post-Increment operatörü operandı işleme koyar ve ardından değerini 1 artırır.
  - Örneğin; a'nın değeri başlangıçta 1 ise, `b = ++a;` işleminde, önce a'nın değeri bir artılıp ardından b'ye atandığı için, hem a, hem de b'nin değeri 2 olurken, `b = a++;` işleminde önce b'nin değeri a'ya eşitlenip ardından a'nın değeri 1 artırıldığı için b'nin değeri 1 olurken a'nın değeri 2 olacaktır.

# Operatörler

## İlişkisel (Relational) operatörler

İlişkisel operatörler, mantıksal ifadelerde ve karşılaştırmalarda kullanılır.

Bir karşılaştırma sonucu elde edilen değeri, mantıksal doğru (true) ya da mantıksal yanlış (false) olabilir.

Operatör	Gösterimi	Örnek
Eşit (mi?)	==	a == b
Eşit değil (mi?)	!=	a != b
Büyük (mü?)	>	a > b
Küçük (mü?)	<	a < b
Büyük veya eşit (mi?)	>=	a >= b
Küçük veya eşit (mi?)	<=	a <= b

# Operatörler

## Mantıksal (Logical) operatörler

Mantıksal operatörler birden çok karşılaştırma işlemini birleştirip tek bir koşul ifadesi haline getirilmesi için kullanılır.

Operatör	Gösterimi	Örnek
ve (and)	&&	<code>a &gt; 0 &amp;&amp; a &lt; 10</code>
veya (or)		<code>( (a &gt; 0)    (a &lt; 10) )</code>
değil (not)	!	<code>!(a &gt; 0)</code>

```
public class LogicOper {  
    public static void main(String[] args) {  
        int a = 0, b = 0;  
        if (a > 10 && b < 55)  
            System.out.println("Koşul sağlanıyor");  
        else  
            System.out.println("Koşul sağlanmıyor"); // a ?  
    }  
}
```

Koşul sağlanmıyor

# Operatörler

## Bit temelli (Bitwise) operatörler

Bit düzeyinde operatörler, değişkenlerin/sabitlerin tuttuğu değerlerin doğrudan ikili kodlarının bitleri üzerinde işlem yaparlar. Örneğin 6 sayısının iki karşılığı 0110'dır. Bu değer, sonuç 4 bit üzerinde kalmak koşuluyla bir sola kaydırılırsa 1100, tümleyeni (complement) alınırsa 1001 olur. Bit düzeyinde operatörler veriyi bit düzeyinde etkilemektedir.

Operatör	Gösterimi	Örnek
&	a & b	AND
	a   b	OR
^	a ^ b	XOR
~	~a	COMPLEMENT
>>	a >> b	RIGHT SHIFT
<<	a << b	LEFT SHIFT



# Operatörler

## Bit temelli (Bitwise) operatörler

```
public class BitWiseOper {  
    public static void main(String[] args) {  
        int a = 5, b = 7, c = 9;  
        System.out.println("a & b    = " + (a & b));  
        System.out.println("a | b    = " + (a | b));  
        System.out.println("a ^ b    = " + (a ^ b));  
        System.out.println(" ~a     = " + (~a));  
        System.out.println(" ~b     = " + (~b));  
        System.out.println(" ~c     = " + (~c));  
        System.out.println("a >> 1  = " + (a >> 1));  
        System.out.println("a >> 2  = " + (a >> 2));  
        System.out.println("a << 1  = " + (a << 1));  
        System.out.println("a << 2  = " + (a << 2));  
        System.out.println("a >>> 2 = " + (a >>> 2));  
    }  
}
```

a & b	=	5
a   b	=	7
a ^ b	=	2
~a	=	-6
~b	=	-8
~c	=	-10
a >> 1	=	2
a >> 2	=	1
a << 1	=	10
a << 2	=	20
a >>> 2	=	1



# Operatörler

## Operatör Öncelikleri

- ✓ En yüksek öncelik (varsa) en içten başlamak koşuluyla dışa doğru parantez içleridir.
- ✓ Aritmetik operatörde çarpma ve bölme operatörünün toplama ve çıkarma operatörüne göre önceliği vardır. Aynı öncelikli operatör soldan sağa sırayla işletilir.
- ✓ Aritmetik operatörler ilişkisel operatöre göre önceliklidir.
- ✓ İlişkisel operatör soldan sağa sırayla işletilir ve mantıksal operatörlere göre önceliklidir.
- ✓ En düşük öncelikli operatör atama operatörüdür.

```
int a = 3, b = 5, c;  
c = a * ++b / 5 - 12 * a + b;           // (15/5)-36+6  
System.out.println("c: " + c);          // -27  
boolean bool = a + b < a * c && a + b >= b + a;  
System.out.println("bool: " + bool);    // bool: false
```

```
int x = 3, y = 5, z;  
z = (x * ++y) / (5 - 12) * (x + y);      // -2*9  
System.out.println("c: " + z);          // -18  
boolean test = (x + y) < (x * z) && (x + y) >= (y + x);  
System.out.println("test: " + test);    // test: false
```